PR #44380 完整报告

vllm-project/vllm

[Bugfix] Fix test_cutlass_moe.py

合并时间：2026-06-05 02:18

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44380

执行摘要

- 一句话：修复 CUTLASS FP8 MoE 测试和 expert_map 传递
- 推荐动作：值得合入。该 PR 修复了长期失效的测试，并修正了一个潜在的功能缺失。建议在合并后监控 CI 中该测试的通过情况。

功能与动机

PR 描述指出测试使用的 FusedMoEConfig 配置不正确 / 不完整，触发了 permute cache 中的断言，导致 test_cutlass_moe.py 几乎全部失败。作者在 issue 评论中附带了 CI 构建日志，确认该测试长时间未通过。同时，fp8 cutlass experts 理应支持外部传入的 expert_map，但代码中硬编码为 None。

实现拆解

1. 修正 cutlass_moe.py 中的 expert_map 传递：在 CutlassExpertsFp8Base.apply 方法中，调用 run_cutlass_moe_fp8 时将之前硬编码的 None 替换为传入的 expert_map 参数，确保外部 expert_map 能够被正确使用。
2. 增强 make_dummy_moe_config 函数（位于 tests/kernels/moe/utils.py）：新增 num_local_experts 和 max_num_tokens 参数，并使其在 FusedMoEConfig 构造中正确传递，避免使用默认的不合理值。
3. 更新测试用例中的配置构造（位于 tests/kernels/moe/test_cutlass_moe.py）：在 run_with_expert_maps 和 run_8_bit 函数中，调用 make_dummy_moe_config 时补全 max_num_tokens、experts_per_token 和 num_local_experts 参数，使其与实际的 token 数和 top-k 值匹配。同时将 global_num_experts 的计算从 moe_tensors.w1_q.shape[0] 改为使用 num_experts 变量，避免 None 问题。
4. 调整 FP8 测试参数传递：在 test_run_cutlass_moe_fp8 中，调用 run_cutlass_moe_fp8 时增加了 expert_map 参数（None），以匹配更新后的函数签名。

关键文件：

- vllm/model_executor/layers/fused_moe/experts/cutlass_moe.py（模块 MoE 专家；类别 source；类型 data-contract；符号 CutlassExpertsFp8Base.apply）：核心文件。修正了 expert_map 参数的传递，从硬编码 None 改为使用传入值，保持对外部 map 的支持。
- tests/kernels/moe/test_cutlass_moe.py（模块 MoE 测试；类别 test；类型 test-coverage；符号 run_with_expert_maps, run_8_bit, test_run_cutlass_moe_fp8）：测试主文件。

修复了配置构造和参数计算，使测试用例能正确通过。

- tests/kernels/moe/utils.py（模块 MoE 工具；类别 test；类型 test-coverage；符号 make_dummy_moe_config）：测试工具文件。扩展了 make_dummy_moe_config 函数，接受并传递关键的配置参数。

关键符号：CutlassExpertsFp8Base.apply, make_dummy_moe_config, run_with_expert_maps, run_8_bit

关键源码片段

vllm/model_executor/layers/fused_moe/experts/cutlass_moe.py

核心文件。修正了 expert_map 参数的传递，从硬编码 None 改为使用传入值，保持对外部 map 的支持。

```
# vllm/model_executor/layers/fused_moe/experts/cutlass_moe.py
# 在 apply 方法中，将 expert_map 参数传递给 run_cutlass_moe_fp8，
# 此前硬编码为 None 导致外部传入的 expert_map 被忽略。

def apply(...):
    ...
    run_cutlass_moe_fp8(
        output,
        hidden_states,
        w1,
        w2,
        topk_ids,
        activation,
        global_num_experts,
        expert_map, # 改为传入参数，而非硬编码 None
        self.w1_scale,
        ...
    )
```

tests/kernels/moe/utils.py

测试工具文件。扩展了 make_dummy_moe_config 函数，接受并传递关键的配置参数。

```
# tests/kernels/moe/utils.py
# 新增 num_local_experts 和 max_num_tokens 参数，
# 使 FusedMoEConfig 构造更准确，避免 permute cache 断言失败。

def make_dummy_moe_config(
    num_experts: int = 1,
    num_local_experts: int | None = None, # 新增参数
    experts_per_token: int = 1,
    hidden_dim: int = 1,
    intermediate_size_per_partition: int = 1,
    in_dtype: torch.dtype = torch.bfloat16,
    max_num_tokens: int = 512, # 新增参数
) -> FusedMoEConfig:
```

```
    return FusedMoEConfig(
        ...
        num_local_experts=num_local_experts
        if num_local_experts is not None
        else num_experts, # 使用传入值
        max_num_tokens=max_num_tokens, # 使用传入值
        ...
    )
```

评论区精华

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更主要集中在测试工具和测试用例的配置修复，以及源码中一个参数的传递方式修正。*expert_map* 参数的传递回归了对外部支持的设计意图，不会引入新问题。测试覆盖了 FP8 MoE 的多种参数组合，风险可控。
- 影响：影响范围限于 CUTLASS FP8 MoE 相关的测试和核心逻辑。修复后，*test_cutlass_moe.py* 将能正常通过，且 *CutlassExpertsFp8Base* 的 *apply* 方法能够正确支持外部 *expert_map*，这对分布式推理的专家并行场景可能有帮助。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR